Paper No. 16

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte RANDALL R. HEISCH

Appeal No. 97-2525
Application 08/291,370[1]

ON BRIEF

Before KRASS, LEE and TORCZON, Administrative Patent Judges.

LEE, Administrative Patent Judge.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134 from the examiner's final rejection of claims 20, 23-25, 28-30, and 33-34. No claim has been allowed.

References relied on by the Examiner

Pettis et al. (Pettis)	5,212,794	May 18, 1993
Levine et al. (Levine)	5,446,876	Aug. 29, 1995

[1] Application for patent filed August 15, 1994.

(filing date Apr. 15, 1994)

The Rejection on Appeal

Claims 20, 23-25, 28-30, and 33-34 stand finally rejected under 35 U.S.C. § 103 as being unpatentable over Pettis and Levine.

The Invention

The invention is directed to a method and apparatus for reordering the instructions within an executable file to optimize execution thereof by a data processing system. The independent claims are claims 20, 25 and 30. Claim 30 is directed to a computer program product including a computer usable medium containing computer readable program code means for reordering instructions within an executable file. The independent claims are reproduced below:

> 20. A method of reordering the instructions within an executable file to optimize execution thereof by a data processing system, said method comprising the steps of:
>
> recording, during execution of said instructions, trace information including address information;
>
> selecting a subset of said instructions which are indicated by said trace information as being executed frequently;
>
> moving each of said selected instructions from their original physical location to a new physical location at the end of the executable file; and

indicating in each one of said original physical locations said new physical location corresponding to said moved instruction.

25. An apparatus for reordering the instructions within an executable file to optimize execution thereof by a data processing system, said apparatus comprising:

means for recording, during execution of said instructions, trace information including address information;

means for selecting a subset of said instructions which are indicated by said trace information as being executed frequently; and

means for moving each of said selected instructions from their original physical location to a new physical location at the end of the executable file; and

means for indicating in each one of said original physical locations said new physical location corresponding to said moved instruction.

30. A computer program product comprising:

a computer usable medium having computer readable program code means embodied therein for reordering the instructions within an executable file to optimize execution thereof by a data processing system, said computer readable program code means comprising:

means for recording, during execution of said instructions, trace information including address information;

means for selecting a subset of said instructions which are indicated by said trace information as being executed frequently; and

> means for moving each of said selected instructions from their original physical location to a new physical location at the end of the executable file; and
>
> means for indicating in each one of said original physical locations said new physical location corresponding to said moved instruction.

DISCUSSION

We reverse.

A reversal of the rejection on appeal should not be construed as an affirmative indication that the appellants' claims are patentable over prior art. We address only the positions and rationale as set forth by the examiner and on which the examiner's rejection of the claims on appeal is based.

The appellant does not make any argument about the manner in which the examiner applied the teachings from Levine. Accordingly, the issues raised in this appeal only concern the examiner's findings concerning what Pettis shows.

All of the appellant's claims require the reordering of the instructions within an executable file. The appellant argues that in Pettis' invention, it is the computer source code rather than the executable file compiled from that source

code that is reordered. We are not persuaded by the appellant's argument.

While Pettis does describe a two-pass procedure over "computer code", the appellant appears to have incorrectly assumed that both passes are over the same computer code, i.e., source code. Pettis describes that in the first pass, the computer source code is compiled into an executable file but nowhere refers to the second pass as being over the "source code" or indicates that the reordered computer code should or needs to be re-compiled. Moreover, computers do not directly execute source code and thus it is not very meaningful to rearrange portions of the source code program in memory. It also appears that however the source code is stored in memory, the same compiler would still produce the same executable file. Pettis nowhere talks about modifying the compiler to produce a reordered executable file based on different storage arrangements of the source code.

Nevertheless, we are persuaded by the appellant's other arguments that Pettis does not disclose either (1) moving each of selected instructions from their original physical location to a new physical location at the end of the executable file,

or (2) indicating in each one of the original physical locations the new physical location corresponding to the moved instruction. How these features operate are illustrated in an embodiment shown in appellant's own Figure 3. Instructions of higher execution frequency are moved to the end of the executable file and in their place are inserted unconditional branches to the new location. It is the examiner's burden to demonstrate that Pettis discloses the appellant's claimed features.

The examiner cites to column 3, line 51 through column 4, line 52 of Pettis as disclosing the moving of selected instructions to the end of the executable file and placing in their original place an indication of the new location. We have read the cited portions of Pettis and do not find anything to support the examiner's determination. The examiner further cites to column 8, lines 14-39 as disclosing the same. We have read those portions of column 8 and again cannot find the alleged disclosure. The examiner has not established where Pettis discloses that the original locations from where selected instructions have been moved are made to contain indications of the new location for the moved

instructions. From the looks of Pettis' Figure 3, it appears that in Pettis the original locations are re-used to store other selected instructions. Note further that according to Pettis (column 3, lines 45-50), after the reordering program size is reduced, whereas according to the appellant's claimed scheme the program size is necessarily enlarged. It has also not been shown that in Pettis the moved instructions are placed at the end of the executable file. Read in light of appellant's specification, we interpret that limitation to mean that the moved instructions are placed subsequent to or following the end of the executable file. It appears that in Pettis the reordering is all done within the same program space. For instance, see Pettis' Figure 3.

As applied by the examiner, Levine does not make up for the deficiencies of Pettis.

For the foregoing reasons, the rejection of claims 20, 23-25, 28-30, and 33-34 cannot be sustained.

CONCLUSION

The rejection of 20, 23-25, 28-30, and 33-34 under 35 U.S.C. § 103 as being unpatentable over Pettis and Levine is reversed.

Appeal No. 97-2525
Application 08/291,370

REVERSED

ERROL A. KRASS Administrative Patent Judge	)	
JAMESON LEE Administrative Patent Judge	)	BOARD OF PATENT APPEALS AND INTERFERENCES
RICHARD TORCZON Administrative Patent Judge	)	

Richard A. Henkler
IBM CORPORATION
Intell Prop Law Department
Internal Zip 4054
11400 Burnet Road
Austin, TX 78758